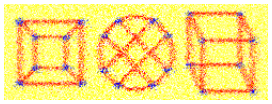

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 16, no. 1, pp. 37–83 (2012)

Lombardi Drawings of Graphs

Christian A. Duncan[1] *David Eppstein*[2] *Michael T. Goodrich*[2]
Stephen G. Kobourov[3] *Martin Nöllenburg*[4]

[1]Dept. of Computer Science, Louisiana Tech Univ.
http://www.latech.edu/~duncan/
[2]Dept. of Computer Science, Univ. of California, Irvine
http://www.ics.uci.edu/~(goodrich|eppstein)/
[3]Dept. of Computer Science, University of Arizona
http://cs.arizona.edu/~kobourov/
[4]Faculty of Informatics, Karlsruhe Institute of Technology
http://i11www.iti.kit.edu/~noellenburg

Abstract

We introduce the notion of Lombardi graph drawings, named after the American abstract artist Mark Lombardi. In these drawings, edges are represented as circular arcs rather than as line segments or polylines, and the vertices have *perfect angular resolution*: the edges are equiangularly spaced around each vertex. We describe algorithms for finding Lombardi drawings of regular graphs, graphs of bounded degeneracy, and certain families of planar graphs.

Submitted:	Reviewed:	Revised:	Accepted:
December 2010	July 2011	August 2011	August 2011
	Final:	Published:	
	September 2011	January 2012	
Article type:		Communicated by:	
Regular paper		U. Brandes and S. Cornelsen	

This research was supported in part by the National Science Foundation under grants CCF-0830403, CCF-0545743, and CCF-1115971, by the Office of Naval Research under MURI grant N00014-08-1-1015, by the Louisiana Board of Regents under PKSFI Grant LEQSF (2007-12)-ENH-PKSFI-PRS-03, and by the German Research Foundation under grant NO 899/1-1.
E-mail addresses: christian.duncan@acm.org (Christian A. Duncan) eppstein@ics.uci.edu (David Eppstein) goodrich@ics.uci.edu (Michael T. Goodrich) kobourov@cs.arizona.edu (Stephen G. Kobourov) noellenburg@kit.edu (Martin Nöllenburg)

1 Introduction

The American artist Mark Lombardi [25] was famous for his drawings of social networks representing conspiracy theories. Lombardi used curved arcs to represent edges, leading to a strong aesthetic quality and high readability. Inspired by this work, we introduce the notion of a *Lombardi drawing* of a graph, in which edges are drawn as circular arcs with *perfect angular resolution*: consecutive edges are equiangularly spaced around each vertex. While not all vertices have perfect angular resolution in Lombardi's work, the equiangular spacing of edges around vertices is clearly one of his aesthetic criteria; see Fig. 1.

Traditional graph drawing methods rarely guarantee perfect angular resolution, but poor edge distribution can nevertheless lead to unreadable drawings. Additionally, while some tools provide options to draw edges as curves, most rely on straight-line edges, and it is known that maintaining good angular resolution can result in exponential drawing area for straight-line drawings of planar graphs [17, 26]. Our requirement of perfect angular resolution forces us to use curved edges, since even very simple graphs such as cycles cannot be drawn with perfect angular resolution and straight edges.

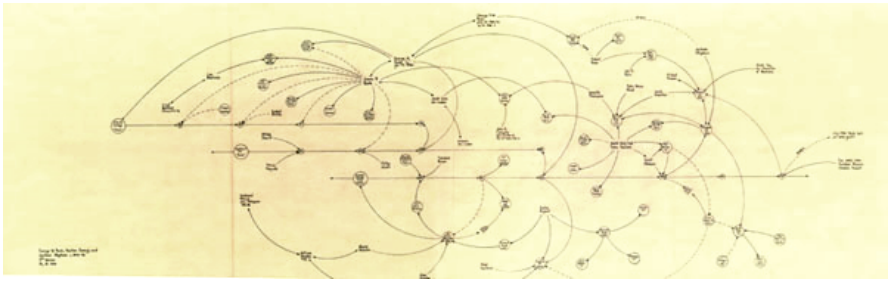


Figure 1: Mark Lombardi, *George W. Bush, Harken Energy, and Jackson Stevens c.1979-90*, 1999. Graphite on paper, $24\,1/8 \times 44\,1/4$ inches [25, cat. no. 19, p. 99].

1.1 New Results

We define a ***Lombardi drawing*** of a graph G to be a drawing of G in the plane in which vertices are represented as points (or as disks or labels centered on those points), edges are represented as line segments or circular arcs between their endpoints, and every vertex has perfect angular resolution, as measured by the angle formed by the tangents to the edges at the vertex. We do not necessarily insist that the drawings are free of crossings; the drawings of Lombardi had crossings, sometimes even in cases where they could have been avoided. We also do not consider crossings when we measure the angular resolution of a drawing. However, we do require that the only vertices that intersect the arc for an edge (u, v) are its two endpoints u and v.

Inspired by the overall circular shape of some of Mark Lombardi's drawings,

we define a ***circular Lombardi drawing*** to be a Lombardi drawing in which the vertices lie on a circle. It is almost equivalent to ask for a Lombardi drawing in which the vertices lie on a straight line, as circles and straight lines can be transformed into each other (preserving circularity of arcs and local angular resolution) by a Möbius transformation; the only difference is that vertices on a circle can be connected by a cycle of edges that lie entirely on the circle while vertices on a line cannot. Similarly, we define a ***k-circular Lombardi drawing*** to be a Lombardi drawing in which the vertices lie on k concentric circles. As can be seen from Fig. 1, Mark Lombardi sometimes used the x-coordinates of vertices to convey extra information such as a timeline, so circular Lombardi drawings (transformed to straighten the circle containing the vertices) may be of interest in graph drawing applications in which an additional dimension such as time is to be visualized.

We provide the following results:

- We characterize the regular graphs that have circular Lombardi drawings, we find efficient algorithms for constructing circular Lombardi drawings of d-regular graphs when $d \quad 2 \pmod 4$, and we show that it is NP-complete to test whether a d-regular graph has a circular Lombardi drawing when $d \quad 2 \pmod 4$.

- We describe methods of finding Lombardi drawings for any 2-degenerate graph (a graph that may be reduced to the empty graph by repeated removal of vertices of degree at most 2) and many but not all 3-degenerate graphs.

- We investigate the graphs that have planar Lombardi drawings. We show that certain subclasses of the planar graphs always have such drawings, but that there exist planar graphs with no planar Lombardi drawing.

- We implement an algorithm for constructing k-circular Lombardi drawings with a high degree of symmetry, and we use it to draw many symmetric graphs. We also implement our algorithms for finding circular Lombardi drawings without the assumption of symmetry.

1.2 Related Work

Most previous work on angular resolution concerns straight-line drawings (e.g., see [10, 17, 26]) or polyline drawings (e.g., see [18, 22]). For instance, Di Battista and Vismara [10] give a nonlinear optimization characterization that can find straight-line drawings of embedded planar graphs with a prescribed assignment of angles if such drawings exist.

The angular resolution of drawings with circular-arc edges was previously studied by Cheng *et al.* [8], who showed that maintaining bounded angular resolution in planar drawings may require exponential area even with circular-arc edges. For drawings with cubic Bézier curves, Brandes *et al.* present a method to realize given angles in tree drawings [5] and a force-directed fixed-position

algorithm avoiding small angular resolution [7]. Brandes, Shubina, and Tamassia [6] rotate optimal angular resolution templates. Aichholzer *et al.* [1] show that, for a given embedded planar triangulation with fixed vertex positions, one can find a circular-arc drawing of the triangulation that maximizes the minimum angular resolution by solving a linear program. Finkel and Tamassia [13] also try to optimize angular resolution using force-directed methods for laying out graphs with curved edges.

Our circular Lombardi drawings use a circular layout of vertices that is already popular in combination with other styles of drawing (e.g., see [3, 16, 32]). However, previous methods for circular layouts draw edges as straight line segments or curves perpendicular to the circle, neither of which leads to good angular resolution.

Efrat *et al.* [12] show that given a fixed placement of the vertices of a planar graph, determining whether the edges can be drawn with circular arcs so that there are no crossings is NP-Complete. They also show that if the choices for each circular arc are exactly the two possible half-circles, then the problem has an efficient polynomial-time algorithm via a reduction to 2-satisfiability.

Any tree may be drawn with straight edges and perfect angular resolution. In a separate paper [11], we study the area requirements for tree drawings with perfect angular resolution. We show that, if the edges around each vertex may be permuted, any tree has a straight-line drawing with perfect angular resolution and polynomial area. However, we provide examples showing that, when the order of the edges is fixed around each vertex, straight-line tree drawings with perfect angular resolution may require exponential area. As we prove in that paper, Lombardi drawings can achieve polynomial area even when the edge ordering around each vertex is fixed.

2 Circular Lombardi Drawings of Regular Graphs

We begin by investigating *circular Lombardi drawings*, Lombardi drawings in which all vertices are placed on a circle. As we show, drawings of this type exist for many regular graphs. Our proofs use the following basic geometric observation illustrated in Figure 2:

Property 1 *Let A be a circular arc or line segment connecting two points p and q that both lie on circle O. Then A makes the same angle to O at p that it makes at q. Moreover, for any p and q on O and any angle 0 , there exists an arc, line segment, or pair of collinear rays A connecting p and q, making angle with O, and lying either inside or outside of O.*

The case of two collinear rays is problematic (we only allow edges to be represented by arcs or line segments) but easily avoided by perturbing the vertices on O.

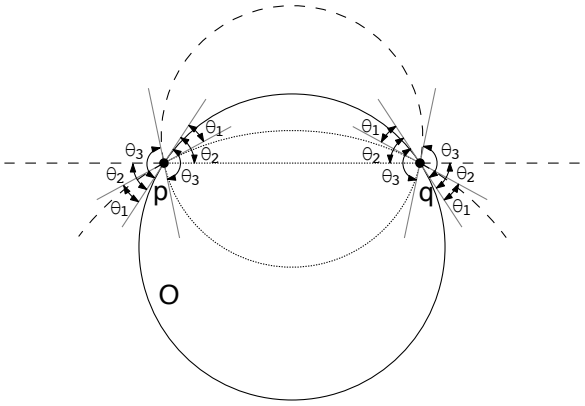


Figure 2: Two points p and q on circle O and the angles formed between O and (degenerate) circular arcs inside (dotted) and outside (dashed) of O.

2.1 Characterization

In this work, we define a ***decomposition*** of a graph G as the set of edge-induced spanning subgraphs of G formed by a partition of the edges in G.

Lemma 1 *A d-regular graph G has a circular Lombardi drawing if and only if G can be decomposed into a disjoint union of 1-regular and 2-regular graphs and one of the following conditions is true: $d \equiv 2 \pmod 4$, one of the 2-regular subgraphs is bipartite, or one of the 2-regular subgraphs is a Hamiltonian cycle.*

Proof: Suppose G has a circular Lombardi drawing on a circle O centered at o and let v be a vertex (of degree d) on O. To describe, from the perspective of o, the rotation of the star formed by the tangent rays of the d circular-arc edges incident to v about v we define the *twist* τ_v of v as the smallest angle between line segment vo and any tangent ray of v. If the smallest angle is found clockwise of vo or if there are two equal smallest angles we assign a positive sign to τ_v, otherwise we assign a negative sign to τ_v; see Figure 3. Observe that $|\tau_v| \leq \pi/d$. If v and w are adjacent in G, then by Property 1 $\tau_v = -\tau_w$ (Fig. 3(b)) except when there are two equal sharpest angles at both v and w, in which case $\tau_v = \tau_w$ (Fig. 3(a)). In each connected component either all vertices have the same twist, and the star of tangent rays is symmetric with respect to reflections through axis vo, or the component is bipartite; all vertices on one side of the bipartition have one twist, and all vertices on the other side of the bipartition have the opposite twist.

We can decompose each connected component of G into 1-regular and 2-regular graphs by partitioning the edges of the component according to the angle they make with circle O. For a bipartite component in which the vertices on the two sides of the bipartition have different twists, this forms a decomposition into 1-regular graphs (some of which may be combined in pairs to form bipartite 2-regular graphs). When $d \equiv 2 \pmod 4$ and a component of G is not bipartite,

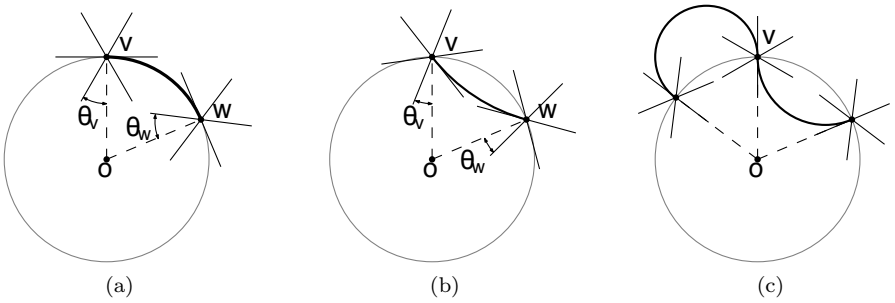
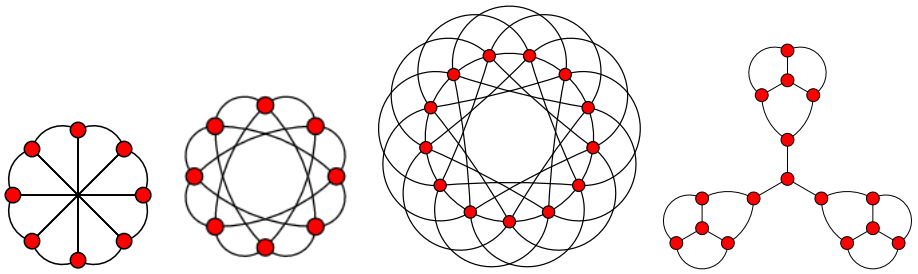


Figure 3: A degree-6 vertex v with twist $_v$ and tangent rays indicated. (a) Vertex v having two equal sharpest angles and v's adjacent vertex w having twist $_w = _v$; (b) The twist $_v$ having positive value and v's adjacent vertex w having twist $_w = -_v$; (c) Vertex v having twist $_v = 0$.

the only possibilities for a symmetric twist are to make some edges parallel or perpendicular to O. Edges that are parallel to O must be drawn as arcs of O through all vertices, so they form a Hamiltonian cycle; see Figure 3(a). Edges perpendicular to O must form even-length cycles that alternate between the inside and outside of O; see Figure 3(c). Thus, in all cases a graph with a circular Lombardi drawing can be decomposed into 1-regular and 2-regular graphs matching the conditions of the lemma.

In the other direction, suppose that G can be decomposed into 1-regular and 2-regular graphs with the additional conditions of the lemma. By combining pairs of 1-regular graphs into a single 2-regular graph, we may assume that all but at most one of these subgraphs are 2-regular. Then we may choose an equiangular set of angles, draw each 2-regular graph as a set of arcs that meet O at one of these fixed angles, and draw the 1-regular graph (if it exists) as a set of arcs that are perpendicular to and interior to O. If d is divisible by four, we can choose these angles in such a way that no angle is parallel to the circle O and no angle is perpendicular to O. If d is odd, the angles can be chosen so that the 1-regular subgraph of G is perpendicular to and interior to O, and all other angles are neither perpendicular nor parallel to O. If $d \quad 2 \pmod 4$ and one of the 2-regular graphs is a Hamiltonian cycle, we may draw it using edges that lie on C, placing the vertices in the order of this cycle. And if $d \quad 2 \pmod 4$ and one of the 2-regular graphs is bipartite, we may draw it using edges that are perpendicular to O, taking care in the vertex placement to avoid using an edge that connects two diametrally opposite points on O via an exterior arc. In both of these cases where $d \quad 2 \pmod 4$ we then draw the other subgraphs of the decomposition using arcs that are neither parallel to nor perpendicular to O. □

Theorem 1 *Every regular graph G of degree divisible by four has a circular Lombardi drawing. A regular graph of odd degree has a circular Lombardi draw-*



(a) (b) (c) (d)

Figure 4: (a) A circular Lombardi drawing of the 3-regular Wagner graph; (b) A circular Lombardi drawing of the 4-regular graph $K_{4;4}$; (c) The 6-regular Paley graph connecting integers modulo 13 if their difference is a quadratic residue; (d) A 3-regular graph that has no perfect matching and therefore has no circular Lombardi drawing.

ing if and only if it has a perfect matching. A regular graph of degree congruent to two modulo four has a circular Lombardi drawing if and only if it is Hamiltonian or has a 2-regular bipartite subgraph.

Proof: This follows from Lemma 1 together with Petersen's theorem that a regular graph of even degree can always be decomposed into 2-regular subgraphs [29, 30]. □

2.2 Algorithms

In the cases of odd degree and degree divisible by four, when a circular Lombardi drawing exists it can be constructed in polynomial time: the decomposition into 1-regular and 2-regular graphs can be found in polynomial time by graph matching techniques, and the remaining steps of our drawing method are straightforward. The matching techniques dominate the running time and can be solved in $O(dn^{1.5})$ time [27].

Figures 4(a–c) show drawings produced by this method for 3-regular, 4-regular, and 6-regular graphs. Figure 4(d) shows a 3-regular graph that does not have a perfect matching, and that therefore has no circular Lombardi drawing.

The following corollaries describe specific running times for computing circular Lombardi drawings of various subclasses of regular graphs. In particular, for bipartite regular graphs of bounded degree and regular graphs with a bounded degree divisible by four, the method of Theorem 1 leads to a linear-time algorithm.

Corollary 1 *Every bipartite d-regular graph has a circular Lombardi drawing that can be constructed in time $O(dn \log d)$.*

Proof: It is known that every bipartite regular graph can be decomposed into

perfect matchings in the given time bound [2, 9, 31].[1] The result follows by applying Theorem 1 to this decomposition. □

Corollary 2 *Every $4k$-regular graph has a circular Lombardi drawing that can be constructed in time $O(kn \log k)$.*

Proof: This is a consequence of the constructive proof for Petersen's theorem that a regular graph of even degree can always be decomposed into 2-regular subgraphs [29, 30]. Let G be any $4k$-regular graph. We first compute an Euler tour [15] of G in $O(kn)$ time. We then construct a bipartite $2k$-regular graph G^0 as follows: for every vertex v $V(G)$, create two vertices v^+ and $v^\square$; and for each edge (u, v) $E(G)$, with the tour visiting the edge from u to v, add an edge $(u^+, v^\square)$ to G^0. We again decompose G^0 into perfect matchings in time $O(kn \log k)$ [2, 9, 31]. Collapsing the two copies of each vertex transforms each perfect matching of G^0 into a 2-regular subgraph of G. Applying Theorem 1 to the decomposition with the condition that the degree is divisible by four yields the stated result. □

Corollary 3 *Every d-regular graph of odd degree having a perfect matching has a circular Lombardi drawing that can be constructed in time $O(dn^{1.5})$.*

Proof: We first compute a perfect matching in $O(dn^{1.5})$ time [27]. The graph formed by removing the matched edges is a regular graph of even degree. As in Corollary 2, we can then apply the technique from Petersen's theorem that a regular graph of even degree can always be decomposed into 2-regular subgraphs [29, 30]. Applying Theorem 1 to the decomposition with the condition that the degree is odd and the graph has a perfect matching yields the stated result. □

Corollary 4 *Every 3-regular bridgeless graph has a circular Lombardi drawing that can be constructed in time $O(n \log^3 n \log \log n)$.*

Proof: The result that every 3-regular bridgeless graph has a perfect matching (equivalently, a decomposition into a 2-regular and a 1-regular subgraph) is known as Petersen's theorem [30]. Such a matching can be found in the stated time bound via an algorithm based on dynamic 2-edge-connectivity testing data structures [4, 21, 33]. □

An implementation of our algorithms for circular Lombardi drawing is described later, in Section 5.1.

2.3 Complexity

Our characterization of d-regular graphs with circular Lombardi drawings completely resolves the computational complexity of finding these drawings when

[1] The fact that every regular bipartite graph has a decomposition into matchings is commonly attributed to König [23] but is equivalent to a result proved in terms of point-line configurations in the 1894 Ph.D. thesis of Ernst Steinitz.

$d \quad 2 \pmod 4$: when d is divisible by 4, Corollary 2 applies, and when d is odd, Corollary 3 applies. Therefore, a circular Lombardi drawing can be found in polynomial time in these cases whenever it exists. The remaining case concerns d-regular graphs for which $d \quad 2 \pmod 4$. However, as the following theorem from Har-Peled [20] shows and whose proof we summarize, testing the existence of a circular Lombardi drawing in this case is NP-complete.

Lemma 2 (Har-Peled [20]) *For any constant $d \quad 3$, it is NP-complete to test whether a given d-regular graph is Hamiltonian. For even d, the problem remains NP-complete for graphs with an odd number of vertices.*

Proof: The problem of testing Hamiltonicity is known to be NP-complete on 3-regular graphs. To reduce the problem of testing Hamiltonicity of a d-regular graph G to the problem of testing Hamiltonicity on $(d+1)$-regular graphs, form a graph G' as the disjoint union of two copies of G, with the two copies of each vertex connected by a gadget formed by removing an edge from the complete graph K_{d+2} and connecting the two endpoints of the removed edge to the two copies.

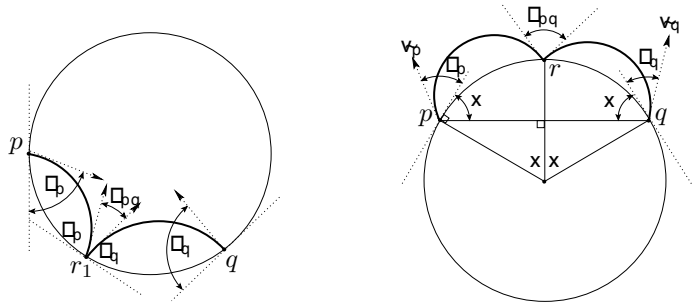
In the graphs produced by this reduction, there are many edges (the edges connecting each copied vertex to the gadget connecting it to the other copy) that must be included in any Hamiltonian cycle. When the degree $(d+1)$ is even, replacing any such edge by a clique (the same gadget used to connect copies of vertices) changes the parity of the number of vertices in the graph, showing that the problem remains NP-complete for regular graphs of even degree with an odd number of vertices. □

Theorem 2 *When $d \quad 2 \pmod 4$, it is NP-complete to test whether a d-regular graph has a circular Lombardi drawing.*

Proof: For d-regular graphs with an odd number of vertices, it is not possible to partition the vertices into even-length cycles, so by Theorem 1 a circular Lombardi drawing exists if and only if the graph is Hamiltonian. The result then follows immediately from Lemma 2. □

3 Two-Degenerate and Three-Degenerate Graphs

The ***degeneracy*** of a graph G is the minimum number d such that G can be reduced to the empty graph by repeatedly removing a vertex of degree at most d; equivalently, it is the minimum degree in the subgraph of G that maximizes the minimum degree [24]. If a graph G has degeneracy at most d, it is known as d-degenerate. In this section we consider algorithms for creating Lombardi drawings, rather than circular Lombardi drawings, of 2-degenerate and 3-degenerate graphs with a specified cyclic ordering of the edges around each vertex. The main idea of these algorithms is to delete a low-degree vertex, draw the remaining graph with the appropriate angles at each of its vertices,



(a) (b)

Figure 5: (a) Angle calculation; (b) Circle construction.

and then find a position for the deleted vertex that allows it to be connected to the drawing of the remaining graph.

For 2-degenerate graphs, as we detail below, when we add back the vertices in reverse order of deletion, there is always a circle on which they can be added so we can choose one point on the circle that is not crossed by a previously drawn feature. However, for 3-degenerate graphs there are two points at which the point can be added to give the correct edge angles (the common intersection points of three circles) so there might be circumstances under which this addition is forced to create an undesirable edge-vertex or vertex-vertex intersection.

The results in this section rely on the following geometric property:

Property 2 *Suppose we are given two points p and q with associated vectors v_p and v_q and an angle α_{pq}. Consider all pairs of circular arcs that leave p and q with tangent vectors v_p and v_q respectively and meet at an angle α_{pq}. The locus of meeting points for these pairs of arcs is a circle.*

Proof: Let r_1 be the meeting point of one such pair of arcs. Let O be the circle defined by the three points p, q, and r_1. From Property 1, the angle α_p that the arc from p makes with O as it leaves p is the same as when it arrives at r_1. Similarly, let α_q be the angle of the arc with O at both q and r_1. Therefore, we know that the angle formed by the intersection of the two arcs at r_1 is $\alpha_{pq} = 2\pi - \alpha_p - \alpha_q$; see Fig. 5(a).

Now, for any other point r_2 on O, a circular arc from p through r_2 with the same outgoing tangent vector v_p must again form the same angle α_p with O at both p and r_2. The same holds for the angle α_q at q and r_2. Therefore, the angle formed by the intersection of the two arcs at r_2 is also α_{pq}.

We can also determine the equation for this circle O. Our goal is to calculate the angle formed by the center of O and the two points p and q. From that, we can use basic trigonometry to calculate the position of the center based on the positions of p and q. For simplicity, assume that the two fixed points p and q are horizontally aligned; see Fig. 5(b). Let r be the point on O halfway between p and q. Since r lies directly above the center of the circle, we know that the

desired angle is exactly $2x$, where x is the angle formed by the horizontal line (from p to q) and the tangent to O at p (or q). From v_{p}, we know the angle, say $_{\mathsf{ph}}$, between the outgoing arc from p and the horizontal line. In Fig. 5(b), this corresponds to the angle $_{\mathsf{p}} + x$. Similarly, we have angle $_{\mathsf{qh}} = _{\mathsf{q}} + x$.

Finally, from above, we know the angle at r is $_{\mathsf{pq}} = \ - _{\mathsf{p}} - _{\mathsf{q}}$. Solving for x yields that $2x = _{\mathsf{ph}} + _{\mathsf{qh}} - _{\mathsf{p}} - _{\mathsf{q}} = _{\mathsf{ph}} + _{\mathsf{qh}} + _{\mathsf{pq}} - \ .$ □

3.1 2-Degenerate Graphs

Theorem 3 *Every 2-degenerate graph with a specified cyclic ordering of the edges around each vertex has a Lombardi drawing.*

Proof: Order the vertices by repeatedly removing a low-degree vertex. Reinsert the vertices in reverse order creating subgraphs $G_0, G_1 \ldots G_{\mathsf{n}}$ with the invariant that after each insertion the drawing is a *partial* Lombardi drawing $_{\mathsf{i}}$ of G_{i} where some vertices may not yet have all of their neighbors placed. To insert a new vertex $v = v_{\mathsf{i}+1}$ with degree two in $G_{\mathsf{i}+1}$ (the case for degree one is simpler) let p and q be its two neighbors in $G_{\mathsf{i}+1}$. Since there is a specified ordering around p, which has already been placed in $_{\mathsf{i}}$, there is a unique tangent vector v_{p} associated with the arc from p to v. Similarly, there is a unique tangent vector v_{q}. In addition, since the degree of v in G is known and the ordering of the neighbors at v is also given, there is a unique angle $_{\mathsf{pq}}$ associated with the two arcs from p and q to v. From Property 2, we may choose to place v at any position on the circle defined by p, q, and $_{\mathsf{pq}}$. Choosing a point v that does not coincide with any other arcs or vertices already placed guarantees we have a valid drawing $_{\mathsf{i}+1}$. □

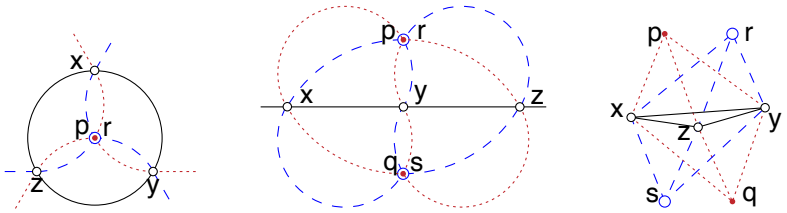
Corollary 5 *Every outerplanar or series-parallel graph has a Lombardi drawing.*

Proof: This follows from the fact that these graphs are 2-degenerate. □

3.2 3-Degenerate Graphs

An algorithm following the same approach can be used to draw many, but not all, 3-degenerate graphs. In this case we have three points p, q, and r that we want to connect by arcs to an unplaced new vertex v. Each pair of known points yields a circle of possible choices for v. These three circles, $O_{\mathsf{pq}}, O_{\mathsf{pr}}, O_{\mathsf{qr}}$, have to pairwise cross, and where they cross the third one must also cross because fixing the angles between two pairs of incoming arcs at the new point fixes all angles. Every graph with maximum degree four is either 4-regular or 3-degenerate, so the same algorithm applies in this case.

However, for certain graphs and certain orderings of the edges around the vertices of the graph, this algorithm can fail by placing a vertex on another edge or vertex. An example in which this occurs is the seven-vertex split graph G_7 formed by adding four independent vertices p, q, r, and s to a triangle xyz, with an edge from each of p, q, r, and s to each of x, y, and z, as shown in



(a) (b) (c)

Figure 6: A 7-vertex 3-degenerate graph that has no Lombardi drawing with the given vertex ordering. (a) A Möbius transformation makes one triangle equilateral, forcing the other 4 vertices to be placed at the centroid and the point at infinity; (b) A different transformation with finite vertex locations; (c) A straight-line drawing of the graph.

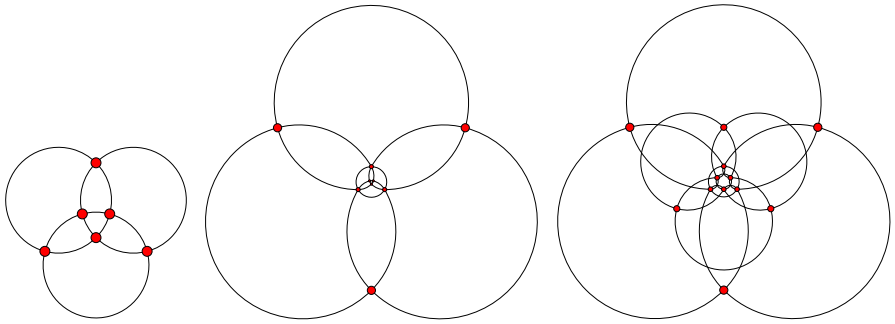
Figure 6. In any Lombardi drawing of G_7 with the edge order as shown, we can assume by making an appropriate Möbius transformation of the drawing that xyz is equilateral. It follows that the only possible locations for p, q, r, and s are the centroid of the equilateral triangle and the point at infinity, so at least two vertices would have to be placed at the same point, forming an invalid drawing.

4 Planar Lombardi Drawings

4.1 Planar Graphs Without Planar Lombardi Drawings

Not every planar graph has a planar Lombardi drawing. To see this, consider the *k-nested triangle graphs*, maximal planar graphs with $3k$ vertices formed by k nested triangles with $k-1$ six-cycles connecting consecutive triangles. A k-nested triangle graph may also be formed geometrically by gluing $k-1$ octahedra end-to-end.

As can be seen in Figure 7, the 2-nested and 3-nested triangle graphs have planar Lombardi drawings. The 4-nested triangle graph, however, does not. If it did have such a drawing, its middle two triangles would form circles (the only smooth curve formed by three circular arcs). By an appropriate Möbius transformation, the outer circle O can be assumed to have its three vertices equally spaced around it. The three circles C_1, C_2, and C_3 that (by Property 2) describe the potential positions of the vertices on the inner circle have the same radius as O and meet at the center of O, and the inner circle would have to be tangent to all three of C_1, C_2, and C_3. However, the only circle tangent to all three is exterior to O, concentric with O and having twice the radius of O. Therefore, using an edge ordering around each vertex that comes from a planar embedding but enforcing perfect angular resolution leads to a nonplanar drawing, shown in Figure 7(c).



(a) $\mathsf{k} = 2$ (b) $\mathsf{k} = 3$ (c) $\mathsf{k} = 4$

Figure 7: k-nested triangle graphs. The 2-nested and 3-nested triangle graphs have planar Lombardi drawings, but the 4-nested triangle graph does not.

4.2 Halin Graphs

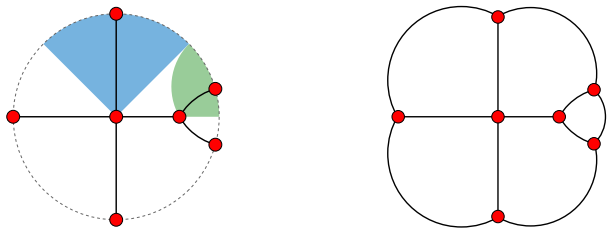
A Halin graph [19] is a planar graph obtained from a plane tree T (with at least four vertices and with no vertices of degree 2), by connecting all the leaves of T into a cycle in the order given by its embedding. As we now describe, Halin graphs (and the graphs formed in the same way from trees with degree-2 vertices) have planar Lombardi drawings that can be constructed using hyperbolic geometry.

We draw T within a Poincaré disk model of the hyperbolic plane, with its leaves on the boundary circle of the model, and then draw the cycle connecting the leaves outside this circle. If T is drawn using hyperbolic line segments, with perfect angular resolution, then its edges will form circular arcs in the Poincaré model; the conformal (angle-preserving) nature of the Poincaré model implies that the angular resolution of the hyperbolic line segments equals the angular resolution of these Euclidean arcs.

For a given straight-line drawing of a rooted tree in the hyperbolic plane, and a non-root vertex v, partition the hyperbolic plane into wedges bounded by the bisectors of the angles around the parent of v and define the ***dominance region*** of v to be the wedge containing v. Equivalently, in a Voronoi diagram generated by the rays from the parent of v to its children, the dominance region of v is the Voronoi cell containing v. We define a ***good hyperbolic drawing*** of a rooted tree T to be a drawing in which the edges are straight line segments or rays in the hyperbolic plane, the leaves are placed on the circle at infinity, and the dominance regions for two vertices v and w are either nested within each other (if one of the two vertices is an ancestor of the other) or disjoint otherwise. Two dominance regions in a good hyperbolic drawing are shown in Figure 8(a).

Lemma 3 *Every rooted tree has a good hyperbolic drawing.*

Proof: We use induction on the number of non-leaf nodes in the given tree T. As a base case, when there is one non-leaf node, it may be placed at the center of the Poincaré disk model of the hyperbolic plane with its leaves at



(a) (b)

Figure 8: (a) A good hyperbolic drawing of a seven-node tree, with the dominance regions of two leaves of the tree shown as shaded regions; (b) The Lombardi drawing formed by adding arcs outside the Poincaré model, at 30° angles to the boundary, connecting consecutive leaves.

the limit points of equally-spaced rays (radii of the disk model). Otherwise, let v be a non-leaf that is as far from the root of T as possible, and let T' be formed from T by removing all children of v. Then by induction, T' has a good hyperbolic drawing. In this drawing, v is on the circle at infinity; let R be the ray connecting the parent of v to v. For any position x along this ray, let $_x$ be the maximum angle made to R by a line that stays within the dominance region of v. Then $_x$ varies continuously along R, starting from a value of $/d$ at the parent of v (where d is the degree of the parent) and ending with a value of at v itself. If the degree of v in T is d', there must be an intermediate position x on R for which $_x = (1-1/d')$. If we move v to x and place its leaf children at the limit points of equally spaced rays around x, the result is a good hyperbolic drawing of T. □

Theorem 4 *Every Halin graph has a planar Lombardi drawing that may be constructed in linear time.*

Proof: Root the tree T at an arbitrarily chosen non-leaf node, and construct a good hyperbolic drawing of T according to Lemma 3. Draw the cycle connecting the leaves of T using circular arcs that meet the circle bounding the Poincaré model at angles of 30° as in Figure 8(b). Then each non-leaf node of T has perfect angular resolution from the tree drawing, and each leaf node has perfect angular resolution because the ray connecting it to its parent in T is perpendicular to the boundary circle and therefore at 120° angles from the two arcs connecting it to adjacent leaves. □

4.3 Other Classes of Planar Graphs

The networks formed by two-dimensional soap bubbles naturally form 3-regular planar Lombardi drawings: they have circular arcs as their edges (the boundaries between bubbles), and 120° angles at each vertex where three arcs meet [28].

However, we do not have a precise characterization of the graphs that can be formed in this way.

The vertices of every Platonic solid, Archimedean solid, and prism lie on a common sphere. In all but two cases (the snub cube and snub dodecahedron) one may draw the edges of the polyhedron as circular arcs on the sphere with perfect angular resolution. By stereographic projection, each of these graphs has a Lombardi drawing in the plane. For instance, Figure 7(a) depicts the graph of the octahedron drawn in this way.

All outerplanar and series-parallel graphs have Lombardi drawings (Corollary 5), but we do not know whether they all have planar Lombardi drawings.

5 Implementations

5.1 Circular Lombardi Drawings

We have implemented in the Python programming language the algorithms of Section 2 for constructing circular Lombardi drawings of regular graphs, as vector graphic images in the SVG format.

Specifying an input as a graph without any differentiation of its edges would be problematic in two respects: first, if we specified the input in this way, our program would need to be able to solve efficiently the NP-complete problem of finding an appropriate partition into 1- and 2-regular spanning subgraphs in the case that the degree is congruent to two modulo four. And second, a single graph may have more than one possible partition of this type, and we desired the choice of partition to be specifiable by the program's operator. To solve both of these problems, we chose to make the input to the program be not just a graph but a partition of the graph into 1- and 2-regular subgraphs.

In more detail, the program takes as input a sequence of command-line arguments, each of which specifies a single 1-regular or 2-regular spanning subgraph of the given graph. The graph itself is then constructed as the union of these subgraphs. Each 1-regular or 2-regular subgraph is specified using LCF notation [14], a format that specifies for each vertex (in clockwise order around the circle on which the vertices lie) the number of positions in the order by which it differs from its neighbor in the subgraph. The pairs of neighbors specified in this way form a directed graph with outdegree one, which we require either to be the orientation of a matching (with two directed edges for each undirected edge) or of a disjoint union of cycles (with one directed edge for each undirected edge). For instance, the eight-vertex cube graph drawn in Figure 9 could be specified in this way by two command-line arguments, the first "3,-3,3,-3,3,-3,3,-3" specifying the pattern of offsets for the inner 1-regular subgraph of the drawing, and the second "1,1,1,1,1,1,1,1" specifying the outer Hamiltonian cycle.

The original application for the LCF format was in the specification of 3-regular Hamiltonian graphs, and in keeping with that application our implementation defaults to including a Hamiltonian cycle as one of its regular spanning subgraphs. For non-Hamiltonian graphs, or graphs in which a Hamiltonian

cycle is not desired as part of the partition into 1- and 2-regular subgraphs, a command-line option negates this default. In addition, a standard refinement of the LCF notation allows for groups of offsets to be matched with their negations and then repeated a given number of times. In this abbreviated form of LCF notation, the same cube drawing could be specified as the single command-line argument "[3,−] 4".

The angles that the edges of the drawing make with the circle on which the vertices lie is determined both by the degree of the vertices and (when the degree is even) by whether the default Hamiltonian cycle is included as part of the graph; if it is included, it is drawn along the circle on which the vertices lie. In graphs with odd degree, the Hamiltonian cycle is drawn with angles as close to this circle as possible. The other 1-regular and 2-regular subgraphs are drawn so that their ordering in the sequence of command line arguments matches their ordering in terms of the angles they make with the circle of vertices, innermost to outermost.

With this format, all regular graphs that have a circular drawing may be specified. Our implementation also has built into it a list of some well known regular graphs and their LCF notations, so that they may be specified by name instead of numerically.

The drawing algorithm itself is very straightforward, consisting only of a trigonometric calculation of the radii of curvature of the circular arcs representing each edge, and conversion of those parameters to SVG objects representing either a circular arc or (when the radius is infinite) a straight line segment. The SVG output of the program is generated using simple print statements.

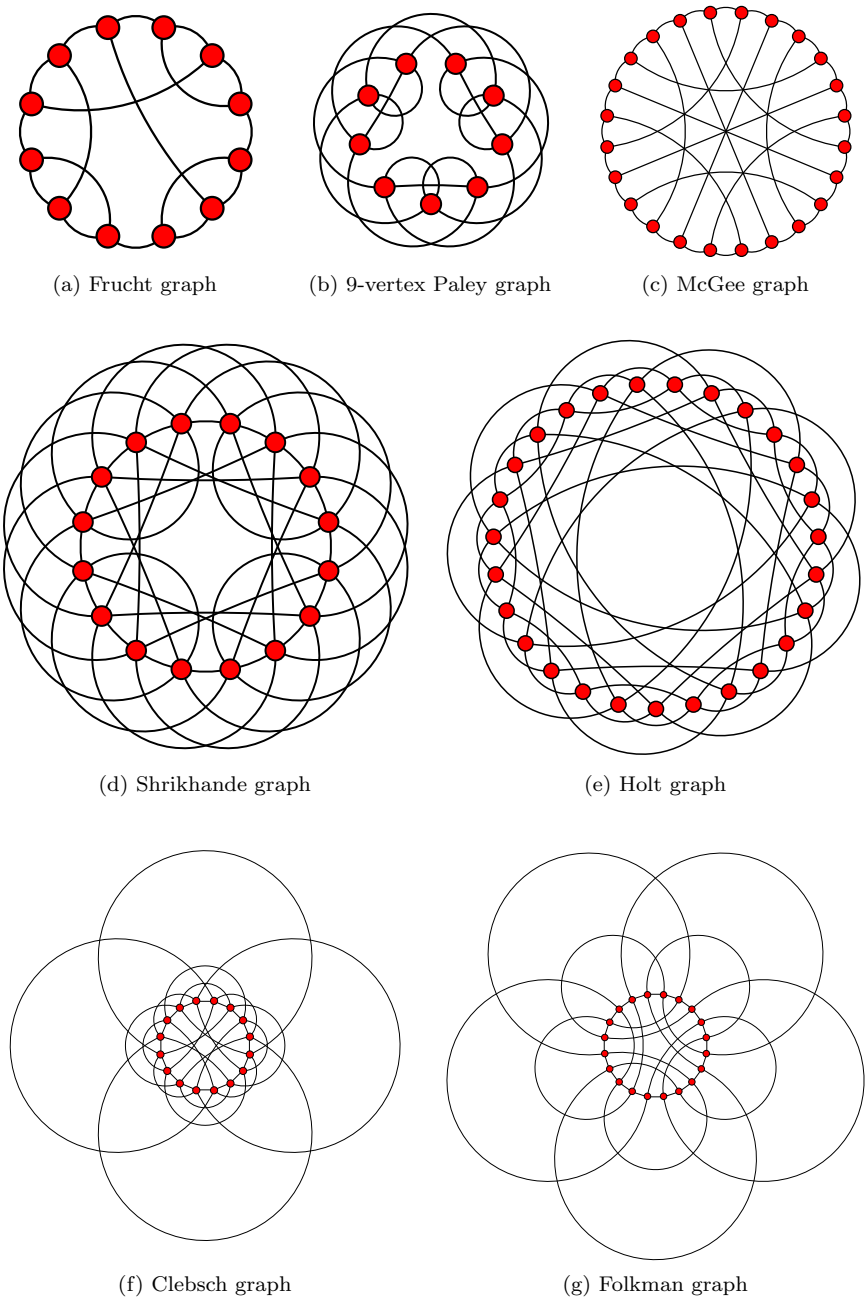
Figures 4 (a–c), 9 and 10 show examples of the output from our implementation. The implementation code can be found online at `http://www.ics.uci.edu/~eppstein/0xDE/ls/CircularLombardi.py`.

5.2 The Lombardi Spirograph

We have also implemented a program for constructing k-circular Lombardi drawings of graphs with dihedral symmetry; we call it the *Lombardi Spirograph*, as its drawings resemble those created by the Spirograph$^{\text{TM}}$ drawing toy produced by Hasbro, Inc.

In the drawings constructed by our program, each vertex can have arbitrarily many neighbors on the same circle, but at most three neighbors on smaller circles. The reason for this is that a circle on which the vertices have two or three inward neighbors has a unique radius for which the vertices have perfect angular resolution, whereas if there were a larger number of inner neighbors it might not be possible to make all connections with perfect angular resolution. The radius for circles on which the vertices have one inner neighbor is not fixed by this connection pattern but is chosen heuristically by our software to achieve a uniform vertex spacing.

As with our program for circular Lombardi drawings, the input to the Lombardi Spirograph is specified as a sequence of command line arguments separated by spaces or dashes. The first argument is a number, the number of vertices

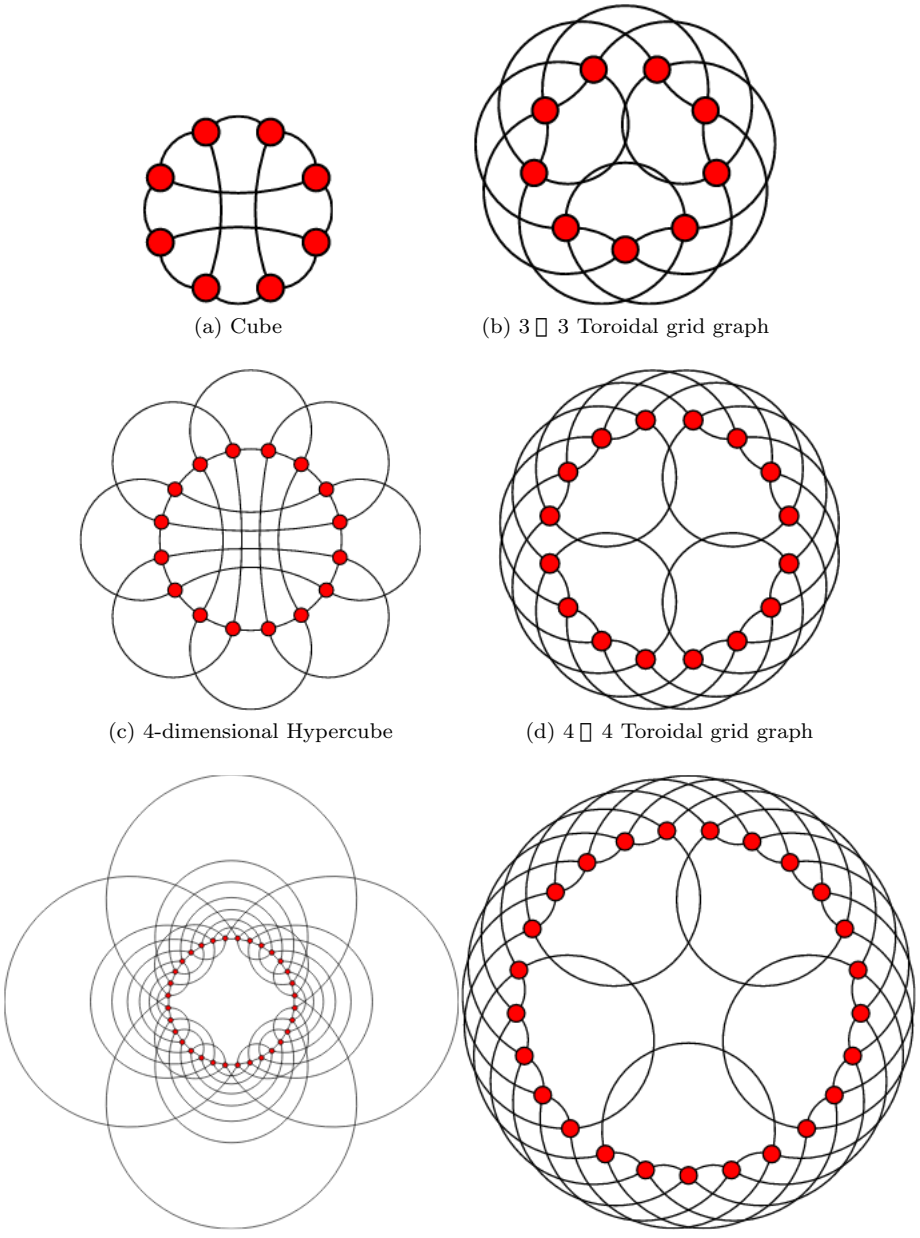


(a) Frucht graph (b) 9-vertex Paley graph (c) McGee graph

(d) Shrikhande graph (e) Holt graph

(f) Clebsch graph (g) Folkman graph

Figure 9: Sample drawings by our circular Lombardi drawing implementation.



(a) Cube

(b) 3 □ 3 Toroidal grid graph

(c) 4-dimensional Hypercube

(d) 4 □ 4 Toroidal grid graph

(e) 5-dimensional Hypercube

(f) 5 □ 5 Toroidal grid graph

Figure 10: Further sample circular Lombardi drawings. The 3×3 Toroidal grid graph is equivalent to the 9-vertex Paley graph. The 4×4 Toroidal grid graph is equivalent to the 4-dimensional Hypercube. Notice how different assignments of the 2-regular subgraphs yield different drawings.

to put on each concentric circle of the drawing; our program is only capable of constructing drawings in which each of these vertices is symmetric to each other vertex on the same circle. Subsequent arguments specify, for each of the circles of the drawing from innermost to outermost (possibly including a degenerate radius-zero circle with one vertex at the center of the drawing) the connection pattern of the vertices on that circle to each other, using letters "a", "b", "c", etc. to specify connections between vertices one position apart on the circle, two positions apart, three positions apart, etc. The ordering of these letters is used to determine the radial order of the edges at each vertex. The command line argument for one of the concentric circles ends with a number (by default, zero) specifying the offset between vertices on the circle and on the next larger circle. The number of circles is determined by the number of arguments.

For instance, the Grötzsch graph in Figure 11(c) (a small nonplanar triangle-free four-chromatic graph) was drawn with this program, using the command line "5-x-1-b". The number 5 indicates that there are five vertices per circle. The "x" indicates that there is a degenerate inner circle with one vertex at the center of the drawing (connected radially to the vertices on the next circle). The "1" indicates that the inner of the two non-degenerate circles has no connections between pairs of vertices on the same circle, and that each vertex on that circle has edges to the two vertices one position clockwise and counterclockwise on the outer circle. The "b" indicates that, on the outer circle, each vertex is connected to the vertices two steps away from it.

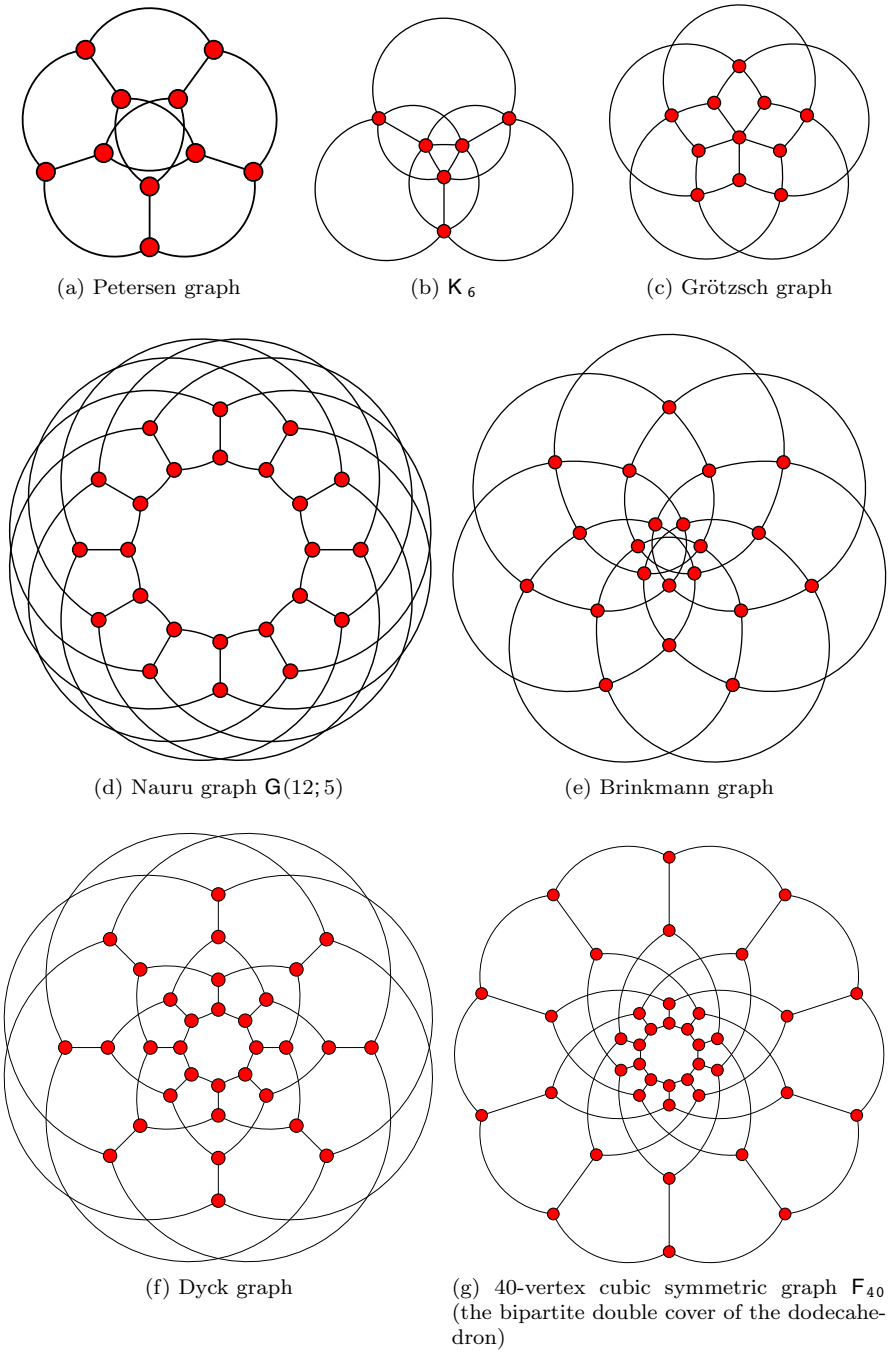
The logic of the program consists of simple case analysis and trigonometric calculations for determining whether edges are curved or straight and, if curved, what their radius of curvature is. A single vertex and its incident edges is generated per circle and is then replicated by rotating it around the origin of the coordinate system using complex-number multiplications. As with our circular drawing program, the output is vector graphics in the SVG format, generated using simple print statements.

Figures 7 (a & b), and 11 were all drawn using this program. The implementation code can be found online at `http://www.ics.uci.edu/~eppstein/0xDE/ls/LombardiSpirograph.py`.

6 Conclusions

We have begun an investigation into Lombardi drawings and found algorithms based on graph matching, incremental construction, hyperbolic geometry, and symmetry display for constructing drawings of this type. Based on our constructions, we can show that many regular graphs, sparse graphs, special classes of planar graphs, and symmetric graphs have Lombardi drawings, and we have found drawings of this type for many well-known graphs. We have implemented our method for constructing circular Lombardi drawings of regular graphs when they exist. In addition, we have implemented a method, called the *Lombardi Spirograph*, for producing Lombardi drawings of graphs with dihedral symmetry.

There are many related problems that remain open, including the following:



(a) Petersen graph

(b) K_6

(c) Grötzsch graph

(d) Nauru graph $\mathsf{G}(12;5)$

(e) Brinkmann graph

(f) Dyck graph

(g) 40-vertex cubic symmetric graph F_{40} (the bipartite double cover of the dodecahedron)

Figure 11: Sample drawings by the Lombardi Spirograph.

1. Is there an effective classification of 3-degenerate graphs according to whether they can or cannot be drawn in a way that avoids overlapping features?

2. Are there efficient methods for producing planar Lombardi drawings for outerplanar graphs, series-parallel graphs, and 3-regular planar graphs?

3. Two-dimensional soap bubbles (partitions of the plane into regions with fixed areas, with minimal total perimeter) form Lombardi drawings of 3-regular 3-connected planar graphs. Which planar graphs can be realized in this way?

It would also be of interest to combine Lombardi drawing with other standard graph drawing quality criteria such as edge-length minimization. In general, we believe that Lombardi drawings will be a fruitful area for much additional research.

References

[1] O. Aichholzer, W. Aigner, F. Aurenhammer, K. Č. Dobiášová, and B. Jüttler. Arc triangulations. *Proc. 26th Eur. Worksh. Comp. Geometry (EuroCG 2010)*, pp. 17–20, 2010.

[2] N. Alon. A simple algorithm for edge-coloring bipartite multigraphs. *Information Processing Letters* 85(6):301–302, 2003, doi:10.1016/S0020-0190(02)00446-5.

[3] M. Baur and U. Brandes. Crossing reduction in circular layouts. *Proc. 30th Int. Worksh. Graph-Theoretic Concepts in Computer Science (WG 2004)*, pp. 332–343. Springer-Verlag, LNCS 3353, 2005, doi:10.1007/978-3-540-30559-0_28.

[4] T. C. Biedl, P. Bose, E. D. Demaine, and A. Lubiw. Efficient algorithms for Petersen's matching theorem. *J. Algorithms* 38(1):110–134, 2001, doi:10.1006/jagm.2000.1132.

[5] U. Brandes and B. Schlieper. Angle and distance constraints on tree drawings. *Proc. 14th Int. Symp. on Graph Drawing (GD 2006)*, pp. 54–65. Springer-Verlag, LNCS 4372, 2007, doi:10.1007/978-3-540-70904-6_7.

[6] U. Brandes, G. Shubina, and R. Tamassia. Improving angular resolution in visualizations of geographic networks. *Data Visualization 2000. Proc. 2nd Eurographics/IEEE TCVG Symp. Visualization (VisSym 2000)*, pp. 23–32. Springer-Verlag, 2000.

[7] U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. *J. Graph Algorithms Appl.* 4(3):135–155, 2000, `http://jgaa.info/accepted/00/BrandesWagner00.4.3.pdf`.

[8] C. C. Cheng, C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Drawing planar graphs with circular arcs. *Discrete Comput. Geom.* 25(3):405–418, 2001, doi:10.1007/s004540010080.

[9] R. Cole, K. Ost, and S. Schirra. Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Combinatorica* 21(1):5–12, 2001, doi:10.1007/s004930170002.

[10] G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM J. Discrete Math.* 9(3):349–359, 1996, doi:10.1137/S0895480194264010.

[11] C. A. Duncan, D. Eppstein, M. T. Goodrich, S. G. Kobourov, and M. Nöllenburg. Drawing trees with perfect angular resolution and polynomial area. *Proc. 18th Int. Symp. on Graph Drawing (GD 2010)*, pp. 183–194. Springer-Verlag, LNCS 6502, 2011, doi:10.1007/978-3-642-18469-7_17, arXiv:1009.0581.

[12] A. Efrat, C. Erten, and S. G. Kobourov. Fixed-location circular arc drawing of planar graphs. *J. Graph Algorithms Appl.* 11(1):145–164, 2007, `http://jgaa.info/accepted/2007/EfratErtenKobourov2007.11.1.pdf`.

[13] B. Finkel and R. Tamassia. Curvilinear graph drawing using the force-directed method. *Proc. 12th Int. Symp. on Graph Drawing (GD 2004)*, pp. 448-453. Springer-Verlag, LNCS 3383, 2005, doi:10.1007/978-3-540-31843-9_46.

[14] R. Frucht. A canonical representation of trivalent Hamiltonian graphs. *Journal of Graph Theory* 1(1):45–60, 1976, doi:10.1002/jgt.3190010111.

[15] H. N. Gabow. Using Euler partitions to edge color bipartite multigraphs. *Int. J. Parallel Programming* 5(4):345–355, 1976, doi:10.1007/BF00998632.

[16] E. R. Gansner and Y. Koren. Improved circular layouts. *Proc. 14th Int. Symp. on Graph Drawing (GD 2006)*, pp. 386–398. Springer-Verlag, LNCS 4372, 2007, doi:10.1007/978-3-540-70904-6_37.

[17] A. Garg and R. Tamassia. Planar drawings and angular resolution: Algorithms and bounds. *Proc. 2nd Ann. European Symp. on Algorithms (ESA 1994)*, pp. 12–23. Springer-Verlag, LNCS 855, 1994, doi:10.1007/BFb0049393.

[18] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. *Proc. 6th Int. Symp. on Graph Drawing (GD 1998)*, pp. 167–182. Springer-Verlag, LNCS 1547, 1998, doi:10.1007/3-540-37623-2_13.

[19] R. Halin. Über simpliziale Zerfällungen beliebiger (endlicher oder unendlicher) Graphen. *Math. Ann.* 156(3):216–225, 1964, doi:10.1007/BF01363288.

[20] S. Har-Peled. Hamiltonicity of k-regular graphs, `http://cstheory.stackexchange.com/questions/1651/hamiltonicity-of-k-regular-graphs/1656`. Unpublished answer to question on cstheory.stackexchange.com, September 25 2010.

[21] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM* 48(4):723–760, 2001, doi:10.1145/502090.502095.

[22] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica* 16(1):4–32, 1996, doi:10.1007/BF02086606.

[23] D. Kőnig. Gráfok és mátrixok. *Matematikai és Fizikai Lapok* 38:116–119, 1931.

[24] D. R. Lick and A. T. White. K-degenerate graphs. *Canad. J. Math.* 22:1082–1096, 1970, http://www.smc.math.ca/cjm/v22/p1082.

[25] M. Lombardi and R. Hobbs. *Mark Lombardi: Global Networks.* Independent Curators, 2003.

[26] S. Malitz and A. Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Math.* 7(2):172–183, 1994, doi:10.1137/S0895480193242931.

[27] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. *Proc. 21st Ann. Symp. on Foundations of Computer Science (SFCS 1980)*, pp. 17–27. IEEE Computer Society, 1980, doi:10.1109/SFCS.1980.12.

[28] F. Morgan. Soap bubbles in $\mathbb{R}^2$ and in surfaces. *Pacific J. Math.* 165(2):347–361, 1994, http://projecteuclid.org/euclid.pjm/1102621620.

[29] H. M. Mulder. Julius Petersen's theory of regular graphs. *Discrete Mathematics* 100(1-3):157–175, 1992, doi:10.1016/0012-365X(92)90639-W.

[30] J. Petersen. Die Theorie der regulären Graphs. *Acta Math.* 15(1):193–220, 1891, doi:10.1007/BF02392606.

[31] A. Schrijver. Bipartite edge coloring in $O(\ m)$ time. *SIAM J. Comput.* 28(3):841–846, 1999, doi:10.1137/S0097539796299266.

[32] J. M. Six and I. G. Tollis. A framework for circular drawings of networks. *Proc. 7th Int. Symp. on Graph Drawing (GD 1999)*, pp. 107–116. Springer-Verlag, LNCS 1731, 1999, doi:10.1007/3-540-46648-7_11.

[33] M. Thorup. Near-optimal fully-dynamic graph connectivity. *Proc. 32nd Ann. ACM Symp. on Theory of Computing (STOC 2000)*, pp. 343–350. ACM, 2000, doi:10.1145/335305.335345.